9. Switch

Program Name: Switch.java Input File: switch.dat

Your boss recently purchased a bunch of chandeliers from a strange company. These chandeliers consist of a single row of n light bulbs. The annoying thing about these chandeliers is that there isn't a single switch that turns all of the bulbs on or off. Instead, they come with k toggle switches, and flipping a switch flips an arbitrary set of bulbs from OFF to ON or ON to OFF i.e. it toggles the lights. Fortunately, it is easy to determine which bulbs each switch flips by turning them on and off one at a time. So, you wrote down what each switch does in the format of a string n long, with each position consisting of a Y if that switch flips that light, and an N if that switch doesn't affect the state of that light. Thus if switch k operates bulb n, it means if when k is flipped, bulb n goes OFF if it was already ON, and goes ON if it was OFF.

Your boss wants to know the minimum number of switch flips it takes to turn all of the lights in each chandelier on, from a starting position of all OFF. Or, if it takes more than 10 switch flips, the boss deems the chandelier "lame" and "too hard to use", so you throw it out. Write him a program that tells him how many flips it takes, or if it is too many or impossible to do, output "TOO HARD!".

Input
The first line of input contains the number of chandeliers to process.
The first line of each chandelier contains two integers: n and k, for $0 < n \leq 100$, $0 < k \leq 20$.
The next k lines each consist of a single string, n characters long, composed of only Y's and N's as described above.

Output
For **each** chandelier, if the minimum number of switch flips required is ≤ 10, output that number; otherwise output the string "TOO HARD!"

Example Input File

```
3
4 3
YNNN
NYNN
NNYY
4 4
YYNN
YNNY
NNYN
YNYN
11 11
YNNNNNNNNNN
NYNNNNNNNNN
NNYNNNNNNNN
NNNYNNNNNNN
NNNNYNNNNNN
NNNNNYNNNNN
NNNNNNYNNNN
NNNNNNNYNNN
NNNNNNNNYNN
NNNNNNNNNYN
NNNNNNNNNNY
```

Example Output to Screen

```
3
3
TOO HARD!
```

Explanation of sample case:
We will represent the string of lights as a 0 or 1 for each light: 0 if the light is off and 1 if it is on.

First case: Start with 0000
Flip switch 1 (YNNN), get 1000
Flip switch 2 (NYNN), get 1100
Flip switch 3 (NNYY), get 1111
All lights are on, only took 3 flips.

Second case: Start with 0000
Flip switch 1 (YYNN), get 1100
Flip switch 2 (YNNY), get 0101
Flip switch 4 (YNYN), get 1111
All lights are on, took 3 flips

Third case: start with 00000000000
Each switch turns exactly one light on, and we need 11 lights on, so there are no 10 flips you can choose that result in all of the lights on, so you output "TOO HARD!"